IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: David A. Luick	§	Confirmation No.: 6101
	§	
	§	Group Art Unit: 2185
Serial No.: 10/670,715	§	
	§	Examiner: Midys Rojas
Filed: 9/25/03	§	
	§	
For: REDUCTION OF CACHE MISS RATES USING SHARED PRIVATE CACHES	§	

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Midys Rojas, or electronically transmitted via EFS-Web, on the date shown below:

May 15, 2007 /Randol W. Read, Reg. No. 43,876/
Date Randol W. Read

APPEAL BRIEF

Applicants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2185 dated December 20, 2006, finally rejecting claims 1-42. The final rejection of claims 1-42 is appealed. This Appeal Brief is believed to be timely since it is electronically transmitted by the extended due date of June 12, 2007, as set by the filing of a Notice of Appeal on March 12, 2007. Please charge the fee of $500.00 for filing this brief to Deposit Account No. 09-0465/ROC920030293US1.

TABLE OF CONTENTS

Real Party in Interest

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-42 are pending in the application. Claims 1-42 were originally presented in the application. Claims 1-42 stand finally rejected as discussed below. The final rejections of claims 1-42 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

Summary of Claimed Subject Matter

Claimed embodiments include methods (see claims 1-12), apparatus (see claims 13-27), computer systems (see claims 28-32) and computer programs stored on computer readable storage media (see claims 33-42) directed to techniques for reducing cache miss rates in cache memory associated to multiple processors. More specifically, a cache re-allocation scheme for cache lines of the private caches may be selectively enabled and implemented based upon a workload or an expected workload for the processors. *See Application*, page 7, lines 1-8; *Abstract*. In one embodiment, cache miss rates are compared to determine whether one or more of the processors have significantly higher cache miss rates than the average cache miss rates within a processor module or overall. If one or more processors have significantly higher cache miss rates, the cache requests are forwarded from those processors to private caches that have lower cache miss rates and have the least recently used cache lines. See *Application*, page 7, lines 6-14. For a description of the physical environment of the invention, see *Application*, p. 10-16, for a description of an exemplary processor module, see *Application*, p. 16-19, and for a description of a method for reducing cache miss rates, see *Application*, p. 19-21.

A. CLAIM 1 - INDEPENDENT

A method for reducing latencies associated with accessing memory for more than one processor, each coupled with an associated private cache. See *Application*, page 3, lines 18-25, FIG. 3. As claimed, the method includes determining cache miss rates of the more than one processors when issuing cache requests against one or more private caches. See *Application*, page 19, line 21 - page 20, line 1, FIG. 3 (steps 310 and 315). The method also includes comparing the cache miss rates of the more than one processors. See *Application*, page 20, line 1 - page 20, line 20, FIG. 3 (steps 320 and 325). The method also includes allocating cache lines from more than one of the private caches to a processor of the more than one processors based upon the difference between the cache miss rate for the processor and the cache miss rates of

other processors. See *Application*, page 20, line 21 - page 21, line 3, FIG. 3 (steps 330 and 335).

B. CLAIM 5 - INDEPENDENT

A method for reducing cache miss rates for more than one processors, wherein the more than one processors couple with private caches. See *Application*, page 3, lines 3 – page 4, line 10, FIG. 3. As claimed, the method includes monitoring the cache miss rates of the more than one processors. See *Application*, page 19, line 21 - page 20, line 1, FIG. 3 (steps 310 and 315). The method also includes comparing the cache miss rates of the more than one processors to determine when a cache miss rate of a first processor associated with a first private cache of the private caches exceeds a threshold cache miss rate for the more than one processors. See *Application*, page 20, line 1 - page 20, line 20, FIG. 3 (steps 320 and 325). The method also includes forwarding a cache request associated with the first processor to a second private cache of the private caches in response to determining the cache miss rate exceeds the threshold cache miss rate. See *Application*, page 20, line 21 - page 21, line 3, FIG. 3 (steps 330 and 335). The method also includes replacing a cache line in the second private cache with a memory line received in response to the cache request. See *Application*, page 21, lines 4-12, FIG. 3 (steps 340, 345, 350 and 355). The method also includes accessing the cache line in response to an instruction from the first processor. See *Application*, page 21, lines 12-14, FIG. 3 (step 360).

C. CLAIM 13 - INDEPENDENT

An apparatus for reducing cache miss rates for more than one processors, wherein the more than one processors couple with private caches. See *Application*, page 4, lines 11-18, page 16, lines 5-6, FIG. 2. As claimed, the apparatus includes a cache miss rate monitor configured to determine the cache miss rates of the more than one processors when issuing cache requests against the private caches. See *Application*, page 12, lines 7-12, page 16, lines 20-26, page 18, lines 4-8, FIG. 1 (item 130), FIG. 2 (item 250). The apparatus also includes a cache miss rate comparator configured to compare the cache miss rates. See *Application*, page 12, lines 13-20, page 18, lines 9-17, FIG. 1 (item 135), FIG. 2 (item 260). The apparatus also includes a

cache request forwarder configured to allocate cache lines from more than one of the private caches to a cache request of a processor of the more than one processors based upon the difference between the cache miss rate for the processor and the cache miss rates of other processors. See *Application*, page 13, lines 24 – page 15, line 21, page 18, line 2 – page 19, line 4, FIG. 1 (item 145), FIG. 2 (item 270).

D. CLAIM 18 - INDEPENDENT

An apparatus adapted to reduce the latency for accessing memory coupled thereto. See *Application*, page 4, lines 11-18, page 16, lines 5-6, FIG. 2. As claimed, the apparatus includes more than one processors to issue cache requests. See *Application*, page 10, line 10 – page 11, line 14, page 16, lines 5-28, FIG. 1 (items 112-115), FIG. 2 (items 210-212). The apparatus also includes more than one private caches, each individually coupled with one of the more than one processors. See *Application*, page 10, lines 11–19, page 17, lines 17-28, FIG. 1 (items 122-125), FIG. 2 (items 240-242). The apparatus also includes a cache miss rate monitor to determine a cache miss rate with each of the more than one processors. See *Application*, page 12, lines 7-12, page 16, lines 20-26, page 18, lines 4-8, FIG. 1 (item 130), FIG. 2 (item 250). The apparatus also includes a cache miss rate comparator to determine when at least one of the cache miss rates exceeds a threshold. See *Application*, page 12, lines 13-20, page 18, lines 9-17, FIG. 1 (item 135), FIG. 2 (item 260). The apparatus also includes a cache request forwarder to forward a cache request from a processor of the more than one processors that is associated with a cache miss rate determined to exceed the threshold, to a private cache of the more than one private caches associated with another processor of the more than one processors. See *Application*, page 13, lines 24 – page 15, line 21, page 18, line 2 – page 19, line 4, FIG. 1 (item 145), FIG. 2 (item 270).

E. CLAIM 28 - INDEPENDENT

A system. See *Application*, page 5, lines 1-10, page 10, lines 10-11, FIG. 1. As claimed, the system includes a processor module comprising a first processor coupled with a first private cache and a second processor coupled with a second private

cache. See *Application*, page 11, lines 3–16, page 16, line 5 – page 19, line 19, FIG. 1 (items 110 and 150), FIG. 2. The system also includes a cache miss rate monitor to count cache misses associated with the first processor and the second processor. See *Application*, page 12, lines 7-12, page 16, lines 20-26, page 18, lines 4-8, FIG. 1 (item 130), FIG. 2 (item 250). The system also includes a cache miss rate comparator to compare the cache misses associated with the first processor against cache misses associated with the second processor. See *Application*, page 12, lines 13-20, page 18, lines 9-17, FIG. 1 (item 135), FIG. 2 (item 260). The system also includes a cache request forwarder to forward cache requests from the first processor to the second private cache when a number of cache misses associated with the first processor, related to the first private cache, exceeds a number of cache misses associated with the second processor. See *Application*, page 13, lines 24 – page 15, line 21, page 18, line 2 – page 19, line 4, FIG. 1 (item 145), FIG. 2 (item 270).

F. CLAIM 33 - INDEPENDENT

A computer readable medium containing a program which, when executed, performs an operation. See *Application*, page 5, lines 11-16, page 9, line 11 – page 10, line 8, FIG. 3. As claimed, the program contained in the computer readable medium includes determining cache miss rates of more than one processors when issuing cache requests against one or more private caches. See *Application*, page 19, line 21 - page 20, line 1, FIG. 3 (steps 310 and 315). The program contained in the computer readable medium also includes comparing the cache miss rates. See *Application*, page 20, line 1 - page 20, line 20, FIG. 3 (steps 320 and 325). The program contained in the computer readable medium also includes allocating cache lines from more than one of the private caches to a processor of the more than one processors based upon a difference between the cache miss rate for the processor and the cache miss rates of other processors. See *Application*, page 20, line 21 - page 21, line 3, FIG. 3 (steps 330 and 335).

G. CLAIM 36 - INDEPENDENT

A computer readable medium containing a program which, when executed, performs an operation. See *Application*, page 5, lines 17-26, page 9, line 11 – page 10, line 8. As claimed, the program contained in the computer readable medium includes monitoring cache miss rates of more than one processors. See *Application*, page 19, line 21 - page 20, line 1, FIG. 3 (steps 310 and 315). The program contained in the computer readable medium also includes comparing the cache miss rates of the more than one processors to determine when a cache miss rate of a first processor associated with a first private cache exceeds a threshold cache miss rate for the more than one processors. See *Application*, page 20, line 1 - page 20, line 20, FIG. 3 (steps 320 and 325). The program contained in the computer readable medium also includes forwarding a cache request associated with the first processor to a second private cache in response to determining the cache miss rate exceeds the threshold cache miss rate. See *Application*, page 20, line 21 - page 21, line 3, FIG. 3 (steps 330 and 335). The program contained in the computer readable medium also includes replacing a cache line in the second private cache with a memory line received in response to the cache request. See *Application*, page 21, lines 4-12, FIG. 3 (steps 340, 345, 350 and 355). The program contained in the computer readable medium also includes accessing the cache line in response to an instruction from the first processor. See *Application*, page 21, lines 12-14, FIG. 3 (step 360).

Grounds of Rejection to be Reviewed on Appeal

1. Rejection of claims 1-42 under 35 U.S.C. 102(e) as being anticipated by *Dean et al.* (U.S. Patent No. 6,604,174, hereinafter, "*Dean*").

ARGUMENTS

Claims 1-42 are not anticipated by *Dean* under 35 U.S.C. § 102(e).

The Applicable Standard

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

In this case, *Dean* does not disclose "each and every element as set forth in the claim." For example, *Dean* does not disclose the monitoring of the cache miss rates of more than one processors in order to allocate cache lines of *multiple private caches to a single processor*, as claimed in claims 1, 5, 13, 18, 28, 33, and 36. Rather, *Dean* is directed to the use of a single cache. For example, *Dean* discloses a "performance based system and method for dynamic allocation of *a unified multiport cache*." *Dean*, Col. 1, lines 8-9.

In the telephone interview on February 2, 2007, the Examiner admitted that *Dean* teaches a single unified multi-port cache. However, in the advisory action dated March 6, 2007, the Examiner again asserts that *Dean* discloses the recited element, stating that "a group of cache ways out of the multi-port cache represents one private cache for each processor."

Applicant respectfully submits, however, that a "group of cache ways" of a single cache is certainly not the same as multiple private caches. In fact, Applicant submits that cache ways are but one set of components that, along with other components,

make up an individual cache. For example, see *Dean* Figure 1, and the accompanying description:

> Cache 130 comprises N write ports 131, N read ports 132, and cache memory 500. There are N ways in memory 500 and Z lines in the cache. *Dean* column 5, lines 54-56.

Applicant respectfully submits that, since cache ways are components of a cache, the two cannot be logically equivalent.

In the advisory action, the Examiner makes a second argument that *Dean* teaches the use of multiple private caches, stating:

> Dean also teaches that another architecture that can be implemented for a multiple processor system is one where each processor has its own private cache (Col. 2, line 46-52); consequently, Dean does teach more than one private cache, one for each processor. *Advisory Action.*

The portion of *Dean* cited by the Examiner states:

> Currently, there are two primary architectures for multiple processors. The first is for each processor to have a local cache; the second is for a secondary cache between the processors and the main memory. The latter type of cache is referred to as a "unified" cache. With a unified cache, a processor requesting data queries the secondary cache over a common memory bus after gaining control of that bus. *Dean,* column 2, lines 46-52.

Following this passage, *Dean* further discusses unified caches, stating:

> This scheme has several drawbacks. First, the main memory bus is shared between the processors, meaning that only a single request can be honored at a time. Also, multiple cache look ups can create a bottleneck. The individual processor may not be able to handle several lines of information at a time. Moreover, the amount of cache space set aside for each processor in these systems is usually fixed.
>
> Thus, the most prevalent unified caches allow only single requests, can have cache access bottlenecks, and have fixed cache space per processor. What is needed is a

> system that solves these problems. *Dean,* column 2, lines 53-63.

The passage cited by the Examiner is in fact provided as part of the background section of *Dean*. The first architecture, where each processor has local cache, is introduced merely as an alternative to a unified cache, and not discussed further by *Dean*. As is clear from the full passage quoted above, the techniques disclosed in *Dean* are presented as an approach to solve the problems associated to a unified cache, as an alternative to the use of multiple private caches. Therefore, Applicant submits that *Dean* does not disclose the monitoring of the cache miss rates of more than one processors in order to allocate cache lines of *multiple private caches to a single processor*, as recited in the present claims.

For the above reasons, Applicant respectfully submits that independent claims 1, 5, 13, 18, 28, 33, and 36, as well as their dependents, are allowable. Accordingly, Applicant respectfully requests the withdrawal of this rejection.

CONCLUSION

The Examiner errs in finding that claims 1-42 are anticipated by *Dean* under 35 U.S.C. § 102(e). Withdrawal of the rejections and allowance of all claims is respectfully requested.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Randol W. Read, Reg. No. 43,876/

Randol W. Read
Registration No. 43,876
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellant

CLAIMS APPENDIX

1. (Original) A method for reducing latencies associated with accessing memory for more than one processors, each coupled with an associated private cache, the method comprising:

determining cache miss rates of the more than one processors when issuing cache requests against one or more private caches;

comparing the cache miss rates of the more than one processors; and

allocating cache lines from more than one of the private caches to a processor of the more than one processors based upon the difference between the cache miss rate for the processor and the cache miss rates of other processors.

2. (Original) The method of claim 1, wherein determining the cache miss rates comprises counting cache misses of each of the more than one processors.

3. (Original) The method of claim 1, wherein allocating cache lines comprises forwarding cache requests from the processor to a private cache associated with another processor.

4. (Original) The method of claim 1, wherein allocating cache lines comprises selectively allocating cache lines based upon a priority associated with a cache request of the processor.

5. (Original) A method for reducing cache miss rates for more than one processors, wherein the more than one processors couple with private caches, the method comprising:

monitoring the cache miss rates of the more than one processors;

comparing the cache miss rates of the more than one processors to determine when a cache miss rate of a first processor associated with a first private cache of the private caches exceeds a threshold cache miss rate for the more than one processors;

forwarding a cache request associated with the first processor to a second private cache of the private caches in response to determining the cache miss rate exceeds the threshold cache miss rate;

replacing a cache line in the second private cache with a memory line received in response to the cache request; and

accessing the cache line in response to an instruction from the first processor.

6. (Original) The method of claim 5, wherein monitoring the cache miss rates comprises counting cache misses after a cold start, warm-up period.

7. (Original) The method of claim 5, wherein comparing the cache miss rates comprises comparing the cache miss rates, the cache miss rates being associated with more than one processor modules.

8. (Original) The method of claim 5, wherein the threshold cache miss rate is based upon an average cache miss rate for the more than one processors.

9. (Original) The method of claim 5, wherein forwarding the cache request comprises selecting the second private cache based upon a least recently used cache line associated with the private caches.

10. (Original) The method of claim 9, wherein selecting the second private cache comprises selecting a least recently used cache line based upon a processor module on which the first processor resides.

11. (Original) The method of claim 5, wherein forwarding the cache request comprises selecting the cache request based upon a priority associated with the cache request.

12. (Original) The method of claim 5, wherein forwarding the cache request is responsive to a software instruction that overrides a result of comparing the cache miss rates to forward the cache request to the second private cache.

13. (Original) An apparatus for reducing cache miss rates for more than one processors, wherein the more than one processors couple with private caches, the apparatus comprising:

a cache miss rate monitor configured to determine the cache miss rates of the more than one processors when issuing cache requests against the private caches;

a cache miss rate comparator configured to compare the cache miss rates; and

a cache request forwarder configured to allocate cache lines from more than one of the private caches to a cache request of a processor of the more than one processors based upon the difference between the cache miss rate for the processor and the cache miss rates of other processors.

14. (Original) The apparatus of claim 13, wherein the cache miss rate monitor comprises a plurality of counters, each configured to count cache misses of a corresponding one of the more than one processors.

15. (Original) The apparatus of claim 13, wherein the cache request forwarder is adaptable to forward cache requests from the processor to a private cache associated with another processor.

16. (Original) The apparatus of claim 13, wherein the cache request forwarder is adapted to selectively allocate cache lines based upon a priority associated with a cache request of the processor.

17. (Original) The apparatus of claim 13, wherein the cache request forwarder comprises a least recently cache line table to determine which cache line to allocate for use with the processor.

18. (Original) An apparatus adapted to reduce the latency for accessing memory coupled thereto, comprising:

more than one processors to issue cache requests;

more than one private caches, each individually coupled with one of the more than one processors;

a cache miss rate monitor to determine a cache miss rate with each of the more than one processors;

a cache miss rate comparator to determine when at least one of the cache miss rates exceeds a threshold; and

a cache request forwarder to forward a cache request from a processor of the more than one processors that is associated with a cache miss rate determined to exceed the threshold, to a private cache of the more than one private caches associated with another processor of the more than one processors.

19. (Original) The apparatus of claim 18, wherein the more than one processors and the more than one private caches reside on more than one processor modules.

20. (Original) The apparatus of claim 18, wherein the cache miss monitor comprises more than one cache miss counter, each coupled with one of the more than one processors, to start a count of cache misses after a cold start warm-up period.

21. (Original) The apparatus of claim 18, wherein the cache miss comparator comprises a rate averager to compare the cache miss rates to determine when the cache miss rate of the processor exceeds an average cache miss rate associated with the more than one processors.

22. (Original) The apparatus of claim 18, wherein the cache request forwarder is responsive to a software instruction to forward cache requests from one of the more than one processors to the private cache.

23. (Original) The apparatus of claim 18, wherein the cache request forwarder is adapted to select the private cache based upon a least recently used cache line associated with the private caches.

24. (Original) The apparatus of claim 23, wherein the cache request forwarder is adapted to select the private cache based upon a processor module on which the private cache resides.

25. (Original) The apparatus of claim 17, wherein the cache request forwarder is adapted to select the cache request based upon a priority associated with the cache request.

26. (Original) The apparatus of claim 17, wherein the cache request forwarder inserts the cache request into a cache request queue for the private cache to store the memory line in the private cache.

27. (Original) The apparatus of claim 26, wherein the cache request forwarder comprises an arbitrator to arbitrate between the cache request and another cache request from another processor of the more than one processors, to forward the cache request to the cache request queue.

28. (Original) A system, the system comprising:

a processor module comprising a first processor coupled with a first private cache and a second processor coupled with a second private cache;

a cache miss rate monitor to count cache misses associated with the first processor and the second processor;

a cache miss rate comparator to compare the cache misses associated with the first processor against cache misses associated with the second processor; and

a cache request forwarder to forward cache requests from the first processor to the second private cache when a number of cache misses associated with the first processor, related to the first private cache, exceeds a number of cache misses associated with the second processor.

29. (Original) The system of claim 28, further comprising a historical use file containing a set of one or more tasks and associated cache miss rate information.

30. (Original) The system of claim 29, further comprising a software application to enable the cache request forwarder to forward the cache requests based upon the difference between the number of cache misses associated with the first processor and the number of cache misses associated with the second processor.

31. (Original) The system of claim 28, wherein the cache request forwarder allocates cache lines of the first private cache and the second private cache based upon the difference between the cache miss rates of the first processor and the second processor.

32. (Original) The system of claim 28, wherein the cache request forwarder forwards cache requests from a first processor module of the more than one processor modules to a second processor module of the more than one processor modules, the second module having a least recently used cache line.

33. (Original) A computer readable medium containing a program which, when executed, performs an operation, comprising:

determining cache miss rates of more than one processors when issuing cache requests against one or more private caches;

comparing the cache miss rates; and

allocating cache lines from more than one of the private caches to a processor of the more than one processors based upon a difference between the cache miss rate for the processor and the cache miss rates of other processors.

34. (Original) The computer readable medium of claim 33, wherein allocating cache lines comprises forwarding cache requests from the processor to a private cache of the private caches, wherein the private cache is associated with another processor.

35. (Original) The computer readable medium of claim 33, wherein allocating cache lines comprises selectively allocating cache lines based upon a priority associated with a cache request of the processor.

36. (Original) A computer readable medium containing a program which, when executed, performs an operation, comprising:

monitoring cache miss rates of more than one processors;

comparing the cache miss rates of the more than one processors to determine when a cache miss rate of a first processor associated with a first private cache exceeds a threshold cache miss rate for the more than one processors;

forwarding a cache request associated with the first processor to a second private cache in response to determining the cache miss rate exceeds the threshold cache miss rate;

replacing a cache line in the second private cache with a memory line received in response to the cache request; and

accessing the cache line in response to an instruction from the first processor.

37. (Original) The computer readable medium of claim 36, wherein comparing the cache miss rates comprises comparing the cache miss rates, the cache miss rates being associated with more than one processor modules.

38. (Original) The computer readable medium of claim 36, wherein the threshold cache miss rate is based upon an average cache miss rate for the more than one processors.

39. (Original) The computer readable medium of claim 36, wherein forwarding the cache request comprises selecting the second private cache based upon a least recently used cache line associated with the private caches.

40. (Original) The computer readable medium of claim 39, wherein selecting the second private cache comprises selecting a least recently used cache line based upon a processor module on which the first processor resides.

41. (Original) The computer readable medium of claim 36, wherein forwarding the cache request comprises selecting the cache request based upon a priority associated with the cache request after the cache request misses in the first private cache.

42. (Original) The computer readable medium of claim 36, wherein forwarding the cache request is responsive to a software instruction that overrides a result of comparing the cache miss rates to forward the cache request to the second private cache.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.